

A Bayesian Machine Learning Method for Sensor Selection and Fusion with Application to On-Board Fault Diagnostics

Niranjan Subrahmanya

*School of Mechanical Engineering
Purdue University
West Lafayette, Indiana 47907,
U.S.A.*

nsubrahm@purdue.edu

Yung C. Shin

*School of Mechanical Engineering
Purdue University
West Lafayette, Indiana 47907,
U.S.A.*

shin@purdue.edu

Peter H. Meckl

*School of Mechanical Engineering
Purdue University
West Lafayette, Indiana 47907,
U.S.A.*

meckl@purdue.edu

Abstract - — In applications like feature-level sensor fusion, the problem of selecting an optimal number of sensors can lead to reduced maintenance costs and the creation of compact online databases for future use. This problem of sensor selection can be reduced to the problem of selecting an optimal set of groups of features during model selection. This is a more complex problem than the problem of feature selection, which has been recognized as a key aspect of statistical model identification. This work proposes a new algorithm based on the use of a recently proposed Bayesian framework for the purpose of selecting groups of features during regression and classification. The hierarchical Bayesian formulation introduces grouping for the parameters of a generalized linear model and the model hyper-parameters are estimated using an empirical Bayes procedure. A novel aspect of the algorithm is its ability to simultaneously perform feature selection within groups to reduce over-fitting of the data. Further, the parameters obtained from this algorithm can be used to obtain a rank-order among the selected sensors. The performance of the algorithm is then tested by using diesel engine data for fault detection (43 variables, 8-classes, 30000 records) and comparing the misclassification rates with a varying number of sensors.

I. INTRODUCTION

Condition monitoring is gaining increased attention today in various fields such as optimization of automated systems [1], maintenance of structures [2] and on-board vehicle diagnostics. The difficulty in obtaining precise mathematical models for many uncertain non-linear industrial systems has led to the increasing role of data-based condition monitoring schemes. With the availability of a wide range of accurate sensors, the main research in condition monitoring is now focused on the processing of information obtained from these sensors. Multi-sensor data fusion seeks to increase accuracy by exploiting complementary information, while at the same time increase reliability by exploiting the redundancy provided by different sensors. Many attempts towards achieving this goal can be found in literature [3-9].

Feature-level sensor fusion involves the extraction of features by processing the raw sensor data using various

signal processing methods and then using these features to develop a suitable model. It is expected that with suitable processing, the features are less noisy and contain more information about the condition of interest. A schematic of feature-level sensor fusion is shown in Figure 1. In applications like on-board diagnostics, it is possible to collect data in real time from the vehicles to monitor their conditions and also to transfer and store this data in a centralized database for future use by other vehicles. In such situations, the selection of a compact set of sensors and feature, which completely represent the fault signature, is important to reduce the amount of data being transmitted as well as to achieve efficient storage. Therefore, in order to design a feature level sensor fusion system, the following problems have to be addressed.

1. Selection of a minimal number of sensors without compromising performance. This reduces the cost of installation and simplifies the maintenance of sensors.
2. Selection of the best subset of features from the selected sensors. This ensures good generalization and also lessens the signal processing and data transfer burden in real time monitoring.
3. Training a model from data to predict the process condition

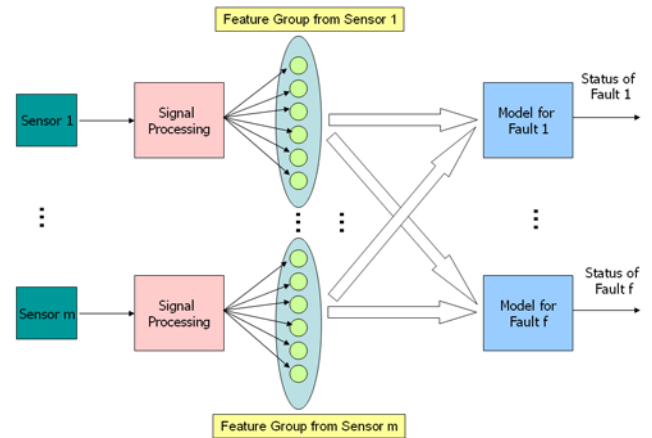


Figure 1. Sensor Selection as Feature Group Selection.

The above figure shows that selecting even a single feature from a particular sensor's feature group to model any of the faults will necessitate selecting the corresponding sensor.

Unfortunately, all the 3 problems stated above are inter-linked as the evaluation of each stage is dependent on other stages. One way to overcome this problem is to group all the features belonging to a sensor together, which reduces the problem of sensor selection to the problem of feature group selection. The problem of feature group selection in fact has a larger scope with potential application in many new fields.

Although a significant amount of research in the area of feature selection may be found in the literature [10], relatively few works on feature group selection are available. While it is possible to come up with certain straightforward extensions of feature selection strategies to feature group selection, they usually do not perform well especially as the number of features in each group becomes large. An extension of the popularly used filter techniques for feature selection based on correlation [10] or mutual information [10] to group feature selection is severely affected in such cases. As a result, recently there have been some attempts made at extending embedded feature selection methods [10, 11] to group feature selection. Specifically, for models which are linear in parameter (models of the form $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$), assuming the feature values to be normalized, the model parameters \mathbf{w} may be considered as scaling factors as each feature is multiplied by the corresponding component of \mathbf{w} to get the term $\mathbf{w}^T \mathbf{x}$, making the magnitude of \mathbf{w}_i indicative of the importance of the i^{th} feature. The essence of embedded selection methods lies in the design of penalty terms for \mathbf{w} , which induce sparsity in terms of groups in the final parameter estimates.

Many such approaches, which are suitable for linear models, have been proposed to date [12-17]. A particular approach that has been gaining popularity lately is the "group lasso", which generalizes the lasso penalty [18] for feature selection to group feature selection. This was originally proposed in [16] as a solution to variable selection for regression when categorical variables also need to be considered. It has since been used for grouping of variables in many applications such as feature selection for multi-output regression [14], micorarray data analysis [19] and for logistic regression [20]. It has been observed that although the group lasso results in setting the weights of many groups to zero, it does not return the smallest possible set of groups that are sufficient to obtain an accurate model [15]. In [21], a modified $\ell_1(\mathbf{w})$ penalty function using the inverse of a pseudo-adjacency graph of feature relations is used to group features together based on their proximity information extracted using SPECT (Single photon emission computed tomography) perfusion imaging [21]. All these methods approach the problem of group feature selection from a regularization point of view. Therefore they require the manual tuning of a trade-off parameter between the regularization term and the error

term. Moreover, only point estimates are obtained after training.

Although fully Bayesian and analytical frameworks have been proposed for automatic feature selection [22, 23], no such attempt has been made for group feature selection. Recently, we proposed a novel model for the problem of feature group selection using a hierarchical Bayesian formulation and gave an algorithm to infer posterior distributions over the parameters and hyper-parameters using variational inferencing [24]. This algorithm brings with it the well known advantages of a fully Bayesian paradigm such as

- Automatic inference of hyper-parameters from data without cross-validation
- Good performance with small data sets
- Probability distributions (and hence confidence intervals) are obtained for the parameters as well as model output instead of point estimates.

In this paper, we follow the same problem formulation but propose a simpler algorithm based on maximizing the log-likelihood with respect to the hyper-parameters of the model. This inference scheme is known by many names such as "type II maximum likelihood" method, or the "evidence for hyper-parameters" method or the "empirical Bayes" method and the reason for maximizing the log-likelihood with respect to the hyper-parameters (rather than the parameters themselves) is the belief that the hyper-parameters cannot over-fit the training data [25]. Thus this method may be considered an extension of the original relevance vector machine (RVM) [22], which was designed to automatically select the most relevant basis vectors for classification and regression using generalized linear models. For group feature selection, the problem involves two stages; the first one being the selection of the most relevant groups followed by the selection the most relevant features from the selected groups. It will be shown that it is possible to incorporate these prior preferences for parameter selection by using additional hyper-parameters in the problem formulation. In this spirit we call the proposed method the Relevant Group Selector (RGS).

Section II presents the new hierarchical formulation of the prior over parameters and provides a discussion on how it reflects the requirements for group feature selection. Section III presents the hyper-parameter estimation algorithm. Results from the application of the proposed framework to the problem of sensor selection and fusion in diesel engine fault diagnostics are presented in Section IV, while the conclusions and scope for utilizing an Engineering Virtual Organization (EVO) for data standardization and software implementation to setup this algorithm are discussed in Section V.

II. HIERARCHICAL BAYESIAN FORMULATION

Given a training set $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{t}_i) \in \mathcal{X} \times \{-1, 1\} : i = 1, \dots, n\}$ for binary

classification or $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{t}_i) \in \mathcal{X} \times \mathbb{R} : i = 1, \dots, n\}$ for regression with $\mathbf{x}_i = [x_{i1}, \dots, x_{ip}]^T \in \mathcal{X} \subset \mathbb{R}^p$, the goal of supervised learning is to learn a function, $y = f(\mathbf{x})$, which not only recalls this information but also generalizes well. Here it is assumed that the function has the structure shown below.

$$y = f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x}) \quad (1)$$

where $\phi(\mathbf{x}) \in \mathbb{R}^d$ is a d -dimensional feature vector extracted from the input \mathbf{x}_i . Assume that the features of the data set are generated by different sensors. Multiple features may be extracted from each sensor. Assume that the d features are extracted from m sensors such that each feature belongs to one and only one sensor. This is not a strict requirement but makes the notation simpler. Let $\mathbf{s}_k \in \mathbb{N}^{d_k}$ denote the feature set of sensor k . Therefore, $\sum_{k=1}^m d_k = d$ and $\mathbf{s}_{k1} \cap \mathbf{s}_{k2} = \emptyset \quad \forall k1 \neq k2$. The goal is to find a good model, which uses features from only a small subset of sensors.

Hierarchical Formulation for Simultaneous Group and Feature Selection

Let $\mathbf{w} = [\mathbf{w}_1, \dots, \mathbf{w}_d]^T \in \mathbb{R}^d$ represent the d -dimensional vector of parameters. A Gaussian prior over the parameters is given by $p(\mathbf{w} | \boldsymbol{\lambda}) = \prod_{j=1}^d (\sqrt{\lambda_j} / \sqrt{2\pi}) \exp(-\lambda_j \mathbf{w}_j^2 / 2)$

where λ_j is the inverse variance for the j^{th} parameter with a higher value of λ_j placing a higher emphasis on sparseness.

As λ_j tends to infinity the probability mass of \mathbf{w}_j concentrates around zero. In order to introduce grouping information based on the sources, \mathbf{w} can be divided into m groups, $\mathbf{w} = [\mathbf{w}^1, \dots, \mathbf{w}^k, \dots, \mathbf{w}^m]$, $\mathbf{w}^k = \{\mathbf{w}_j : j \in \mathbf{s}_k\}$ and each group could have a Gaussian prior controlled by a different parameter from the set $\boldsymbol{\lambda} = \{\lambda_k : k = 1, \dots, m\}$. However, this does not entirely reflect our requirements for simultaneous group and feature selection. Specifically, it does not allow features belonging to the same group to have different prior variances, which is essential for the framework to perform feature selection within the group. In order to achieve this, a flexible prior structure with a larger number of hyper-parameters is considered. Let λ_{1_j} and λ_{2_k} be two hyper-parameters that determine the inverse variance of a normal prior over parameter \mathbf{w}_j , which belongs to the k^{th} group, as

$\mathbf{w}_j \sim N(\mathbf{w}_j | 0, (\lambda_{1_j} + \lambda_{2_k})^{-1})$. Here, a separate hyper-parameter, λ_{1_j} , is assigned to each feature while λ_{2_k} is constrained to be the same for all features belonging to the k^{th} group. Therefore, $\lambda_1 \in \mathbb{R}^d$ and $\lambda_2 \in \mathbb{R}^m$. This structure suitably reflects the belief that the selection of a feature, which is equivalent to having a non-zero value for the

corresponding parameter, is dependent on the relevance of the feature to the prediction task (as estimated by λ_{1_j}) and the relevance of the sensor to which the feature belongs (as estimated by λ_{2_k}). This is because the actual variance of the prior is given by $(\lambda_{1_j} + \lambda_{2_k})^{-1}$ and as either λ_{1_j} or λ_{2_k} tends to infinity this variance tends to zero and the parameter weight is fixed at zero. An infinite value for λ_{1_j} (λ_{2_k}) indicates that the corresponding feature (sensor) is irrelevant. Moreover, when λ_{2_k} tends to zero, indicating that the sensor is almost definitely relevant and selected, we can see that the variance of the parameter prior tends to $\lambda_{1_j}^{-1}$, which results in a prior structure for pure feature selection as in the RVM [22]. Assuming suitable conjugate hyper-priors for λ_{1_j} and λ_{2_k} , the overall prior and hyper-prior structure is as shown below.

$$\begin{aligned} \text{Prior: } p(\mathbf{w} | \boldsymbol{\lambda}) &= \prod_{k=1}^m \prod_{j \in \mathbf{s}_k} N(\mathbf{w}_j | 0, (\lambda_{1_j} + \lambda_{2_k})^{-1}) \\ &= \prod_{k=1}^m \prod_{j \in \mathbf{s}_k} \frac{\sqrt{(\lambda_{1_j} + \lambda_{2_k})}}{\sqrt{2\pi}} \exp\left(-\frac{(\lambda_{1_j} + \lambda_{2_k}) \mathbf{w}_j^2}{2}\right) \end{aligned} \quad (2)$$

$$\begin{aligned} \text{Hyper Prior: } p(\lambda_{1_j} | a_1, b) &= \Gamma(\lambda_{1_j} | a_1, b) \\ &= \frac{b^{a_1} \lambda_{1_j}^{a_1-1} \exp(-b\lambda_{1_j})}{\Gamma(a_1)}, \lambda_{1_j} > 0 \\ p(\lambda_{2_k} | a_2, b) &= \Gamma(\lambda_{2_k} | a_2, b) \\ &= \frac{b^{a_2} \lambda_{2_k}^{a_2-1} \exp(-b\lambda_{2_k})}{\Gamma(a_2)}, \lambda_{2_k} > 0 \end{aligned} \quad (3)$$

III. RELEVANT GROUP SELECTOR (RGS)

The algorithm for regression is presented first. The modifications required to be made to apply the algorithm for classification will be pointed out later. Assuming that observations are generated by the model and corrupted by independent identically distributed Gaussian noise with variance τ^{-1} , the conditional distribution of the target variables, $\mathbf{t} = \{\mathbf{t}_1, \dots, \mathbf{t}_n\}$, given the input, is

$$P(\mathbf{t} | \mathbf{X}, \mathbf{w}, \tau) = \prod_{i=1}^n N(\mathbf{t}_i | \mathbf{w}^T \phi(\mathbf{x}_i), \tau^{-1}) \quad (4)$$

A conjugate hyper-prior is also assumed for the inverse noise variance.

$$P(\tau) = \Gamma(\tau | c, d) \quad (5)$$

Combining the likelihood function (4) and (5) with the weight prior (2) and hyper-prior (3) the complete model specification is obtained. All the hyper-priors can be effectively made non-informative by choosing low values for the parameters of the gamma distribution ($a_1 = a_2 = b = c = d = 10^{-6}$). In an ideal Bayesian framework, we are

interested in predicting the distribution for a new input given the available data.

$$P(\mathbf{t}_{new} | \mathbf{t}) = \int P(\mathbf{t}_{new} | \mathbf{w}, \lambda_1, \lambda_2, \tau) P(\mathbf{w}, \lambda_1, \lambda_2, \tau | \mathbf{t}) d\mathbf{w} d\lambda_1 d\lambda_2 d\tau \quad (6)$$

It is not possible to obtain an exact solution for

$$P(\mathbf{w}, \lambda_1, \lambda_2, \tau | \mathbf{t}) = \frac{P(\mathbf{t} | \mathbf{w}, \lambda_1, \lambda_2, \tau) P(\mathbf{w}, \lambda_1, \lambda_2, \tau)}{P(\mathbf{t})}$$

analytically because of the intractability of computing the normalizing integral to obtain $P(\mathbf{t})$. Hence a decomposition of the posterior probability is made as shown below.

$$P(\mathbf{w}, \lambda_1, \lambda_2, \tau | \mathbf{t}) = P(\mathbf{w} | \lambda_1, \lambda_2, \tau, \mathbf{t}) P(\lambda_1, \lambda_2, \tau | \mathbf{t}) \quad (7)$$

$P(\mathbf{w} | \lambda_1, \lambda_2, \tau, \mathbf{t})$ can be calculated easily because of the Gaussian nature of the prior and the likelihood.

$$P(\mathbf{w} | \lambda_1, \lambda_2, \tau, \mathbf{t}) = (2\pi)^{-(d+1)/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{w} - \boldsymbol{\mu})\right\} \quad (8)$$

$$\text{where } \Sigma = (\tau \Phi^T \Phi + \mathbf{A})^{-1} \quad (9)$$

$$\boldsymbol{\mu} = \tau \Sigma \Phi^T \mathbf{t} \quad (10)$$

with $\Phi = [\phi(\mathbf{x}_1) \ \phi(\mathbf{x}_2) \ \dots \ \phi(\mathbf{x}_n)]^T$ and

$\mathbf{A} = \text{diag}(\lambda_1 + \lambda_2)$, i.e., it is a diagonal matrix with the elements of the vector $\lambda_1 + \lambda_2$ along its diagonal. The term $\lambda_1 + \lambda_2$ is used here to denote a vector where λ_{2k} is added to each λ_{1j} for which $j \in \mathbf{s}_k$. $P(\lambda_1, \lambda_2, \tau | \mathbf{t})$ on the other hand is not as easy to evaluate and in the empirical Bayes procedure, it is replaced by a delta function at its mode, i.e., it is approximated by $\delta(\lambda_1^*, \lambda_2^*, \tau^* | \mathbf{t})$. With this approximation, we hope that

$$\begin{aligned} P(\mathbf{t}_{new} | \mathbf{t}) &= \int P(\mathbf{t}_{new} | \mathbf{w}, \lambda_1, \lambda_2, \tau) P(\mathbf{w}, \lambda_1, \lambda_2, \tau | \mathbf{t}) d\mathbf{w} d\lambda_1 d\lambda_2 d\tau \\ &= \int P(\mathbf{t}_{new} | \mathbf{w}, \lambda_1, \lambda_2, \tau) P(\mathbf{w} | \lambda_1, \lambda_2, \tau, \mathbf{t}) P(\lambda_1, \lambda_2, \tau | \mathbf{t}) d\mathbf{w} d\lambda_1 d\lambda_2 d\tau \\ &= \int \left(\int P(\mathbf{t}_{new} | \mathbf{w}, \lambda_1, \lambda_2, \tau) P(\mathbf{w} | \lambda_1, \lambda_2, \tau, \mathbf{t}) d\mathbf{w} \right) P(\lambda_1, \lambda_2, \tau | \mathbf{t}) d\lambda_1 d\lambda_2 d\tau \\ &\approx \int \left(\int P(\mathbf{t}_{new} | \mathbf{w}, \lambda_1, \lambda_2, \tau) P(\mathbf{w} | \lambda_1, \lambda_2, \tau, \mathbf{t}) d\mathbf{w} \right) \delta(\lambda_1^*, \lambda_2^*, \tau^* | \mathbf{t}) d\lambda_1 d\lambda_2 d\tau \end{aligned}$$

The inner integral, which marginalizes over \mathbf{w} can still be computed analytically since it is the convolution of two Gaussians. Therefore, the main goal now is to find the mode of $P(\lambda_1, \lambda_2, \tau | \mathbf{t})$. This can be done using the expression below.

$$P(\lambda_1, \lambda_2, \tau | \mathbf{t}) \propto P(\mathbf{t} | \lambda_1, \lambda_2, \tau) P(\lambda_1) P(\lambda_2) P(\tau) \quad (11)$$

where

$$\begin{aligned} P(\mathbf{t} | \lambda_1, \lambda_2, \tau) &= \int P(\mathbf{t} | \mathbf{w}, \tau) P(\mathbf{w} | \lambda_1, \lambda_2) d\mathbf{w} \\ &= (2\pi)^{-n/2} |\tau \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T|^{-1/2} \exp\left\{-\frac{1}{2} \mathbf{t}^T (\tau \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T)^{-1} \mathbf{t}\right\} \end{aligned} \quad (12)$$

In order to find the mode of $P(\lambda_1, \lambda_2, \tau | \mathbf{t})$ we maximize the log of this quantity which is given by,

$$\begin{aligned} L &= \log P(\lambda_1, \lambda_2, \tau | \mathbf{t}) \\ &= -\frac{1}{2} \log |\tau \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T| - \frac{1}{2} \mathbf{t}^T (\tau \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T)^{-1} \mathbf{t} \\ &\quad + \sum_{j=1}^n (a_1 \log \lambda_{1j} - b \lambda_{1j}) + \sum_{k=1}^m (a_2 \log \lambda_{2k} - b \lambda_{2k}) + c \log \tau - d\tau \\ &\quad + \text{Constant} \end{aligned} \quad (13)$$

The above quantity can be rewritten in terms of $\lambda_1, \lambda_2, \tau$ and $\boldsymbol{\mu}$ as

$$\begin{aligned} L(\lambda_1, \lambda_2, \tau, \boldsymbol{\mu}) &= -\frac{1}{2} \log |\tau \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T| - \tau \|\mathbf{t} - \Phi \boldsymbol{\mu}\|^2 \\ &\quad - \boldsymbol{\mu}^T \mathbf{A} \boldsymbol{\mu} + \sum_{j=1}^n (a_1 \log \lambda_{1j} - b \lambda_{1j}) \\ &\quad + \sum_{k=1}^m (a_2 \log \lambda_{2k} - b \lambda_{2k}) + c \log \tau - d\tau + \text{Constant} \end{aligned} \quad (14)$$

Setting the partial derivative of $L(\lambda_1, \lambda_2, \tau, \boldsymbol{\mu})$ with respect to each of $(\lambda_1, \lambda_2, \tau, \boldsymbol{\mu})$ to zero and solving each set of equations in an iterative fashion gives the required method for finding the mode of $P(\lambda_1, \lambda_2, \tau | \mathbf{t})$. Once the mode is obtained, $P(\mathbf{w} | \lambda_1^*, \lambda_2^*, \tau^*, \mathbf{t})$ can then be evaluated using (8), (9) and (10) and let Σ^* and $\boldsymbol{\mu}^*$ be the covariance and mean weight matrices obtained using $\lambda_1^*, \lambda_2^*, \tau^*$. The predictions for new inputs are then made as

$$\begin{aligned} P(\mathbf{t}_{new} | \mathbf{t}) &\approx \int P(\mathbf{t}_{new} | \mathbf{w}, \tau) N(\mathbf{w} | \boldsymbol{\mu}^*, \Sigma^*) d\mathbf{w} \\ &= N(\mathbf{t}_{new} | \boldsymbol{\mu}^{*T} \phi(\mathbf{x}_{new}), \sigma^2) \end{aligned} \quad (15)$$

$$\sigma^2 = \frac{1}{\tau^*} + \phi(\mathbf{x}_{new})^T \Sigma^* \phi(\mathbf{x}_{new}) \quad (16)$$

For classification, the conditional distribution of the targets is given by

$$\begin{aligned} P(\mathbf{t} | \mathbf{w}) &= \prod_{i=1}^n \left(1 + e^{-\mathbf{t}_i \mathbf{w}^T \phi(\mathbf{x}_i)}\right)^{-1} = \prod_{i=1}^n \sigma(\mathbf{t}_i \mathbf{w}^T \phi(\mathbf{x}_i)) \\ &= \prod_{i=1}^n \sigma(\mathbf{z}_i) \end{aligned} \quad (17)$$

where $\sigma(z) = (1 + e^{-z})^{-1}$ and $\mathbf{z}_i = \mathbf{t}_i \mathbf{w}^T \phi(\mathbf{x}_i)$. Since the prior over \mathbf{w} is not conjugate to this likelihood function, it is no longer possible to evaluate $P(\mathbf{w} | \lambda_1, \lambda_2, \mathbf{t})$ analytically. In order to overcome this problem, Tipping [22] makes use of a local Gaussian approximation based on Laplace's Method to the posterior distribution of weights. This is done by numerically optimizing for $P(\mathbf{w} | \lambda_1, \lambda_2, \mathbf{t}) \propto P(\mathbf{t} | \mathbf{w}) P(\mathbf{w} | \lambda_1, \lambda_2)$ to find $\boldsymbol{\mu}^*$ and then calculating Σ^* as

$$\Sigma^* = \left(\nabla_{\mathbf{w}} \nabla_{\mathbf{w}} \log P(\mathbf{w} | \lambda_1, \lambda_2, \mathbf{t}) \Big|_{\boldsymbol{\mu}^*} \right)^{-1} = \left(\Phi^T \mathbf{B} \Phi + \mathbf{A} \right)^{-1} \quad (18)$$

where $\mathbf{B} = \text{diag}\{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n\}$ and $\mathbf{B}_i = \sigma(\boldsymbol{\mu}^{*T} \phi(\mathbf{x}_i)) [1 - \sigma(\boldsymbol{\mu}^{*T} \phi(\mathbf{x}_i))]$.

The prediction for new data can be obtained by using the posterior mean weights in (17). Therefore, in order to complete the algorithm, it is necessary to obtain the expressions for the partial derivative of $L(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \tau, \boldsymbol{\mu})$ with respect to each of $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \tau, \boldsymbol{\mu})$. It is generally more convenient to obtain the partial derivative of $L(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \tau, \boldsymbol{\mu})$ with respect to the log of the hyper-parameters $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \tau)$ and setting this to zero is equivalent from the perspective of maximizing L . The partial derivatives are presented below without derivations, which have been skipped to conserve space.

$$\frac{\partial L}{\partial \log \boldsymbol{\lambda}_j} = \frac{\boldsymbol{\lambda}_{1j}}{2(\boldsymbol{\lambda}_{1j} + \boldsymbol{\lambda}_{2k})} - \frac{\boldsymbol{\lambda}_{1j} \boldsymbol{\Sigma}_{jj}^*}{2} - \frac{\boldsymbol{\lambda}_{1j} \boldsymbol{\mu}_j^{*2}}{2} + a_1 - b \boldsymbol{\lambda}_{1j} \quad (19)$$

$$\frac{\partial L}{\partial \log \boldsymbol{\lambda}_{2k}} = \sum_{j \in s_k} \left(\frac{\boldsymbol{\lambda}_{2k}}{2(\boldsymbol{\lambda}_{1j} + \boldsymbol{\lambda}_{2k})} - \frac{\boldsymbol{\lambda}_{2k} \boldsymbol{\Sigma}_{jj}^*}{2} - \frac{\boldsymbol{\lambda}_{2k} \boldsymbol{\mu}_j^{*2}}{2} \right) + a_2 - b \boldsymbol{\lambda}_{2k} \quad (20)$$

$$\frac{\partial L}{\partial \log \tau} = \frac{1}{2} (n - \tau (\mathbf{t} - \Phi \boldsymbol{\mu})^T (\mathbf{t} - \Phi \boldsymbol{\mu}) - \text{trace}(\boldsymbol{\Sigma}^* \Phi^T \Phi)) + c - d \tau \quad (21)$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}} = -2\tau \Phi^T (\mathbf{t} - \Phi \boldsymbol{\mu}) + 2\mathbf{A} \boldsymbol{\mu} \quad (22)$$

Each of the above expressions can be set to zero and solved for the corresponding variables except for (20), which has to be solved numerically (a one-dimensional binary search technique is used to locate the root). From an implementation perspective, it is essential to solve (20) first as given in the algorithm below. The final algorithm which makes use of these solutions is given below.

Algorithm 1. Relevant Group Selection

Initialization: Set $\boldsymbol{\lambda}_{1j}, \boldsymbol{\lambda}_{2k}$ values to 0.5. Set $\boldsymbol{\mu}$ to maximum likelihood solution.

Set τ for regression.

repeat (until parameters converge)

Solve (20) numerically using binary search and update $\boldsymbol{\lambda}_{2k}$ for $k=1, 2, \dots, m$.

Update $\boldsymbol{\lambda}_{1j}$ values to equal the positive root of quadratic expression (19) for $j=1, 2, \dots, d$.

Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using (9) and (10)

$$\tau = \frac{n - \text{trace}(\boldsymbol{\Sigma} \Phi^T \Phi) + 2c}{(\mathbf{t} - \Phi \boldsymbol{\mu})^T (\mathbf{t} - \Phi \boldsymbol{\mu}) + 2d}$$

end
return $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \tau, \boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$

Finally, although the entire explanation has been provided for the case of a single output, it is easy to consider multi-class classification by using a one-versus-all [26] approach and grouping all the parameters corresponding to features from a single sensor together.

IV. EXPERIMENTAL RESULTS

The above algorithm is applied on a real world data set acquired from a set of 16 cylinder diesel engines used in mining applications. The data includes 30000 samples acquired from forty three temperature and pressure sensors connected to engine control units and sampled at 1 Hz at various operating points. The list of variables monitored on the engine is given in Table 1. The engine is assumed to be in eight different states, the first one being healthy and classes two to eight denoting various kinds of faults such as high crankcase pressure, low oil pressure, high intake manifold temperature and so on. More details about the sensors and data acquisition procedure may be found in [27].

TABLE 1 LIST OF VARIABLES MONITORED ON DIESEL ENGINE

Variable	Description	Units
Y	Class Number. 1 – Healthy 2 to 8 – Faulty	-
u_1	Brake Horse Power	Bhp
u_2	ECM Temperature	C (F)
u_3	% Acceleration Pedal	%
u_4	Instantaneous Engine Load	%
u_5	Oil Filter Differential Pressure	kPa (psig)
u_6	Post Filter Oil Pressure	kPa (psig)
u_7	Pre Filter Oil Pressure	kPa (psig)
u_8	Oil Rifle Pressure	kPa (psig)
$u_9 - u_{10}$	LB/RB Boost Pressures	kPa (psig)
$u_{11} - u_{14}$	All Banks IMT	C (F)
u_{15}	Coolant Pressure	kPa (psig)
u_{16}	Coolant Temperature	C (F)
u_{17}	Rail Pressure	kPa (psig)
u_{18}	Battery Voltage	V
u_{19}	LBR Compressor Inlet Temperature	C (F)
$u_{20} - u_{35}$	EGT for 16-Cylinders	C (F)
u_{36}	Avg. Exhaust Temperature	C (F)
u_{37}	Engine Oil Temperature	C (F)
u_{38}	Engine Speed	Rpm
u_{39}	Timing Pressure	kPa (psig)
u_{40}	Fuel Temperature	C (F)
$u_{41} - u_{42}$	LB/RB Avg. Exhaust Temperatures	C (F)
u_{43}	Crankcase Pressure	in of H ₂ O

Y – Output, u – Inputs, LB – Left Bank, RB – Right Bank, F – Front, R – Rear, IMT – Intake Manifold Temperature, EGT – Exhaust Gas Temperature

The performance of RGS is compared to two existing methods: the group lasso [16] and the RVM. The group

lasso has been used in many recent works and requires the tuning of a tradeoff parameter between a penalty term of the form $\sum_{k=1}^m \sqrt{\sum_{j \in S_k} \mathbf{w}_j^2}$ and the training error. The RVM is used

as a baseline measure as it only considers feature selection without grouping information. In addition to this the results from [28] based on using an information theoretic criterion are also presented for comparison purposes.

Four features were extracted from each sensor signal namely the signal itself, the square of the signal, the cube of the signal and the logarithm of the signal. This allows the classification boundary to be slightly non-linear without imposing a significant computational burden for this large dataset. The data set was standardized so that each feature has zero mean and unit standard deviation and a bootstrap sample size of 167, with 180 data points being held out in each sample, was used to estimate the performance of the different algorithms (this is the procedure recommended and followed in [28] based on the data collection procedure). The results are given in the table below. The RGS outperforms the Group Lasso both in terms of classification accuracy and the number of sensors selected. The difference in the prediction accuracy is not very high since very few features were extracted from each sensor and the total number of features is anyhow small compared to the large number of training data points available. The RVM has a slightly better prediction accuracy but at the cost of using a significantly larger number of sensors. Therefore considering the overall tradeoff between the cost of sensor installation and maintenance and the prediction accuracy, the results from RGS could be considered “optimal”.

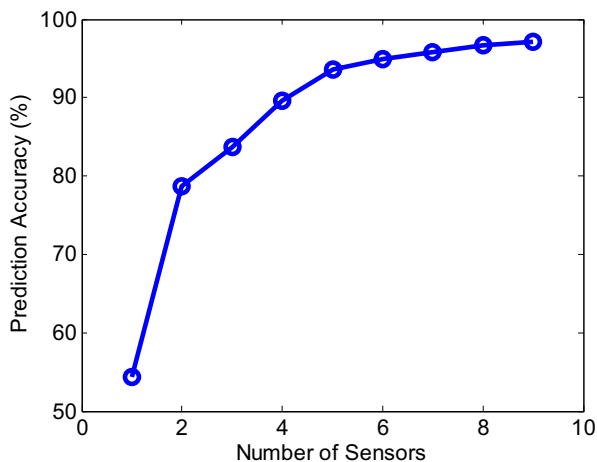


Figure 2. Prediction accuracy versus number of sensors selected according to the ranking returned by RGS

TABLE 2 GROUP SELECTION RESULTS FOR DIESEL ENGINE FAULT DETECTION (THE TABLE PRESENTS RESULTS OBTAINED USING 167 BOOTSTRAP SAMPLES)

RGS		Group Lasso		RVM	
Accuracy (%)	Groups	Accuracy (%)	Groups	Accuracy (%)	Groups
97.11	9	96.79	15	98.32	21

In the next part of the analysis, the rank-order of sensors is determined using the sum of the weights of the features belonging to each sensor, i.e., by sorting the sensor

according to $Weight(k) = \sum_{c=1}^f \sum_{j \in S_k} |\mathbf{w}_j^c|$, where f is the number

of outputs. From Table 2 it can be seen that the RGS has already narrowed down the number of sensors to 9. The ranking among these sensors, in decreasing order of importance, was found to be – 19, 2, 16, 11, 29, 40, 27, 18 and 10 (these correspond to the input number from Table 1). It is now possible to check the performance of the diagnostic system using only the desired top sensors by running the RGS again on this restricted data set. The results obtained are plotted in Fig 2. A prediction accuracy of 89.66% was obtained with the top 4 sensors and a prediction accuracy of 96.68% was obtained with the top 8 sensors. Comparing this to the best accuracy of 89.7% with 8 sensors reported in [28], it can be seen that performing simultaneous sensor selection and model development is indeed beneficial to improving the performance of the system. The simpler linear in parameter model chosen here should also be easier to implement and update for on-board diagnostics. The top 8 sensors selected in [28] are almost entirely different (except for sensor 19), which could also explain the significant difference in prediction accuracy with 9 sensors obtained by the RGS.

V. DISCUSSION ON THE ISSUES RELATED TO CIML VIRTUAL ORGANIZATION

Although the paper has so far focused on the application of the proposed algorithm to engine diagnostics, it is easy to see that the generic nature of the algorithm itself makes it applicable to a broad range of problems where grouping of parameters is of interest. For example, the algorithm can be directly applied to other important problems such as sensor selection in wireless sensor networks, pathway selection in microarray data analysis (where groups of genes have predefined roles), multi-task compressive sensing, region (group of pixels) selection in imaging applications, bandwidth selection in spectroscopic applications and so on. An EVO would be an ideal place for experts from different application communities to come together and discuss the potential uses of this tool to solve problems in their specific areas.

In order to create such a tool, which is readily applicable to different domains, the software development aspects should be given significant consideration, especially for modules designed around the proposed algorithm, which have to be tailored to take into account the idiosyncrasies of individual application domains. Specifically, it is important to address the following issues.

- *Standardization of input data format.* This should include input regarding grouping of parameters.
- *Creation of a centralized database.* It should be possible for different users of a specific domain to upload data to this data-based and also exploit knowledge gained from data already existing in the data-base.
- *Security and validation of input data.* Only authorized users should be allowed access to the data-base and even then the submitted data has to be validated.
- *Real time communication.* The system should also allow the direct uploading of data from units in the field and in turn provide updated models in real time.
- *Interface design.* The Bayesian paradigm allows the extraction of a large amount of information such as the parameter estimates and uncertainty, prediction uncertainty, sufficiency of currently available data, trade-off between using additional features/groups and model accuracy and so on. It is essential to display all the above information in a nice graphical format without overwhelming the non-expert user.
- *Algorithm implementation.* For large scale systems, it might be necessary to consider efficient implementations on the server using parallel programming techniques.
- *Knowledge validation.* While the proposed tool can be used to extract significant correlations and relationships from the database, it is important to have the semantics of these relationships validated by experts in the field before accepting them as process knowledge.

Discussion among the members of the EVO could provide significant inputs regarding the above issues as well as some new ones.

VI. CONCLUSIONS

This work presents a simple algorithm based on the empirical Bayes method for estimating the hyper-parameters in a parameter-free, fully Bayesian framework for simultaneous sensor and feature selection. The algorithm presented here is simpler to implement than the Variational Bayesian algorithm presented in [24]. It is also applicable in various other fields like bio-informatics and spectroscopic analysis where grouping of inputs during data mining is useful. The experimental results presented in this

paper show that the proposed framework effectively manages to achieve its goal of selecting a sparse number of feature groups while simultaneously learning a model that generalizes well. Moreover, additional information regarding the rank-order of the sensors can be attained by ranking the sensors according to the sum of the weights of the features belonging to each sensor. This information can then be used to further reduce the number of sensors while sacrificing performance accuracy only slightly. The reduction in the number of sensors and features will prove to be useful for remotely monitoring systems as it reduces the data transfer and on-line processing required to convey information about the health of the system. It will also help in the creation of compact databases and allow the timely updating of models using a small set of parameters. The performance of the algorithm with only 4 sensors selected this way is still comparable to the results reported in [28].

References

- [1] S. Y. Liang, R. L. Hecker, and R. G. Landers, "Machining Process Monitoring and Control: The State-of-the-Art," *Journal of Manufacturing Science and Engineering*, vol. 126, no. 2, pp. 297-310, 2004.
- [2] K. Worden, and J. M. Dulieu-Barton, "An Overview of Intelligent Fault Detection in Systems and Structures," *Structural Health Monitoring*, vol. 3, no. 1, pp. 85-98, March 1, 2004, 2004.
- [3] Y. Peng, "Intelligent condition monitoring using fuzzy inductive learning," *Journal of Intelligent Manufacturing*, vol. 15, no. 3, pp. 373-380, 2004.
- [4] M. Azam, K. Pattipati, and A. Patterson-Hine, "Optimal sensor allocation for fault detection and isolation," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. pp. 1309-1314.
- [5] L. B. Jack, and A. K. Nandi, "Genetic algorithms for feature selection in machine condition monitoring with vibration signals," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 147, no. 3, pp. 205-212, 2000.
- [6] Q. Liu, X. Chen, and N. Gindy, "Fuzzy pattern recognition of AE signals for grinding burn," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7-8, pp. 811-818, 2005.
- [7] C. Giraud, and B. Jouvencel, "Sensor selection in a fusion process: A fuzzy approach," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. pp. 599-606.
- [8] H. R. Berenji, J. Ametha, and D. Vengerov, "Inductive learning for fault diagnosis," *IEEE International Conference on Fuzzy Systems*. pp. 726-731.
- [9] L. Wang, E. Kannatey-Asibu Jr., and M. G. Mehrabi, "A method for sensor selection in reconfigurable process monitoring," *Journal of Manufacturing*

- Science and Engineering, Transactions of the ASME*, vol. 125, no. 1, pp. 95-99, 2003.
- [10] I. Guyon, S. Gunn, M. Nikravesh *et al.*, *Feature Extraction, Foundations and Applications*, New York: Physica-Verlag, Springer, 2006.
- [11] I. Guyon, and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, March 2003, 2003.
- [12] T. N. Lal, M. Schroder, T. Hinterberger *et al.*, "Support vector channel selection in BCI," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1003-1010, 2004.
- [13] Y. Kim, J. Kim, and Y. Kim, "Blockwise Sparse Regression," *Statistica Sinica*, vol. 16, pp. 375-390, 2006.
- [14] T. Similä, and J. Tikka, "Input selection and shrinkage in multiresponse linear regression," *Computational Statistics and Data Analysis*, vol. 52, pp. 406-422, 2007.
- [15] L. Wang, G. Chen, and H. Li, "Group SCAD regression analysis for microarray time course gene expression data," *Bioinformatics*, vol. 23, no. 12, pp. 1486-1494, 2007.
- [16] M. Yuan, and Y. B. Lin, "Model selection and estimation in regression with grouped variables," *Journal of Royal Statistical Society*, vol. 68, pp. 49-67, 2006.
- [17] P. Zhao, G. Rocha, and B. Yu, "Grouped and hierarchical model selection through composite absolute penalties," *Technical Report, University of California.*, 2006.
- [18] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B*, vol. 58, no. 1, pp. 267-288, 1996.
- [19] S. Ma, X. Song, and J. Huang, "Supervised group Lasso with applications to microarray data analysis," *Bioinformatics*, vol. 8, no. 60, 2007.
- [20] L. Meier, S. van de Geer, and P. Bühlmann, "The group lasso for logistic regression," *Technical Report, Eidgenössische Technische Hochschule.*, 2006.
- [21] J. Stoeckel, and G. Fung, "SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information." p. 8 pp.
- [22] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, vol. 1, no. 3, pp. 211-244, 2001.
- [23] C. M. Bishop, and M. E. Tipping, "Variational Relevance Vector Machines," *Uncertainty in Artificial Intelligence*, C. Boutilier and M. Goldszmidt, eds., pp. 46-53: Morgan Kaufmann, 2000.
- [24] N. Subrahmanya, and Y. C. Shin, "A Variational Bayesian Framework for Group Feature Selection," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. (submitted), 2008.
- [25] D. J. C. McKay, "Bayesian Interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415-447, 1992.
- [26] B. Scholkopf, and A. J. Smola, *Learning with Kernels*, Cambridge, MA: MIT Press, 2002.
- [27] A. A. Joshi, P. H. Meckl, G. B. King *et al.*, "Information-Theoretic Sensor Subset Selection: Application to Signal-Based Fault Isolation in Diesel Engines," in Proceedings of IMECE2006, Chicago, Illinois, USA, 2006.
- [28] A. A. Joshi, S. M. James, P. H. Meckl *et al.*, "Information-Theoretic Feature Selection for Classification," in Proceedings of the 2007 American Control Conference, New York City, USA, 2007.